Amendments to the Claims

1. (currently amended) A method for playing frames of a video adaptively, comprising the steps of:

measuring a spatial frequency of pixel within frames of the video;

measuring a temporal velocity of corresponding pixels between frames of the video;

multiplying the spatial frequency by the temporal velocity to obtain a measure of visual complexity of the frames of the video;

playing the frames of the video at a ~~speed~~ frame rate that corresponds to the visual complexity.

2. (original) The method of claim 1 wherein the video is compressed.

3. (original) The method of claim 2 wherein the spatial frequency is measured from discrete cosine transform coefficients of the pixels in the frames, and the temporal velocity is measured from motion vectors of corresponding pixels between the frames.

4. (original) The method of claim 3 wherein basis functions of the discrete cosine transformation are in a form

$$\cos\left(\frac{\pi k_x(2x+1)}{2N}\right)\cdot\cos\left(\frac{\pi k_y(2y+1)}{2N}\right) = \cos\left(2\pi\frac{k_x}{2N}x + 2\pi\frac{k}{4N}\right)\cdot\cos\left(2\pi\frac{k_y}{2N}y + 2\pi\frac{k}{4N}\right),$$

where k_x is a frequency f_x in an x direction and k_y is a frequency f_y in a y direction in the frame represented as

$$\cos(2\pi\frac{f_x}{N}x + 2\pi\frac{f_y}{N}y),$$

where N is 8 for DCT macro-blocks.

5. (currently amended) The method of ~~claim 5~~ claim 4 wherein each basis function is a superimposition of two 2D sinusoids, one with a spatial frequency $\vec{\mathbf{f}}_1 = (\frac{k_x}{2}, \frac{k_y}{2})$ and another with a spatial frequency $\vec{\mathbf{f}}_2 = (\frac{k_x}{2}, -\frac{k_y}{2})$.

6. (original) The method of claim 5 wherein a particular motion vector is $\vec{\mathbf{v}} = (v_x, v_y)$.

7. (original) The method of claim 6 wherein the visual complexity resulting from the discrete cosine coefficient and the motion vectors are

$$\omega_1 = \vec{\mathbf{f}}_1 \cdot \vec{\mathbf{v}}_1 = \frac{k_x}{2}v_x + \frac{k_y}{2}v_y, \text{ and}$$

$$\omega_2 = \vec{\mathbf{f}}_2 \cdot \vec{\mathbf{v}}_2 = \frac{k_x}{2}v_x - \frac{k_y}{2}v_y.$$

8. (original) The method of claim 3 further comprising:

discarding motion vectors with a low texture;

median filtering the motion vectors; and

fitting a global motion model to the motion vectors to reduce spurious motion vectors.

9. (original) The method of claim 3 wherein the compressed video includes I-frames and P-frames, and further comprising:

determined discrete cosine transformation coefficients of the P-frames by applying motion compensation; and

determining motion vectors for the I-frames by interpolating the motion vectors of the P-frames.

10. (original) The method of claim 1 further comprising:

averaging the visual complexity over a set of frames to determine a complexity of a video segment.

11. (original) The method of claim 1 further comprising:

applying motion blur while plying the video to reduce aliasing.

12. (currently amended) The method of claim 1 wherein ~~a speed~~ the frame rate of playing is inversely proportional to the visual complexity.

13. (original) The method of claim 1 further comprising:

applying coring to spatial filter the video while playing.

14. (original) The method of claim 1 wherein the video is uncompressed.

15. (original) The method of claim 1, in which a temporal distortion of the video is minimized during playback.

16. (original) The method of claim 15, in which the minimizing uses a quantization of the visual complexity.

17. (original) The method of claim 15, in which the minimizing uses a smoothing and filtering of the visual complexity.

18. (original) The method of claim 15, in which the minimizing constructs a piece-wise linear approximation of the visual complexity so that the visual complexity is substantially linear.

19. (original) The method of claim 15, in which the minimizing assigns a constant visual complexity to a consistent temporal segment of the video.